**Amendments to and Listing of the Claims:**

This listing of claims replaces all prior versions and listings of claims in this application.


1. (Currently Amended) A method of replicating data associated with a plurality of transactions in a replication system including a plurality of nodes connected via communication media in a topology, each node including a database, at least some of the nodes being able to independently receive and post transactions, each transaction being one or more transaction steps or transaction operations, including a commit operation, the method comprising:

    (a) initiating and performing ~~transactions~~ transaction steps or transaction operations of a transaction to be executed in a database at an originating node; and

    (b) replicating the ~~transactions~~ transaction steps or transaction operations to at least one or more other nodes while the transaction steps or transaction operations are being performed in the database at the originating node by:

        (i) pausing the commit operation for each transaction being executed in the database at the originating node ~~prior to a commit operation for the transaction,~~

        (ii) assigning a ready to commit token to the transaction,

        (iii) sending the ready to commit token to the one or more other nodes,

        (iv) determining at the one or more other nodes whether the respective databases are prepared for a commit operation for the transaction corresponding to the ready to commit token, and, if so, sending back the ready to commit token to the originating node, and

        (v) executing a commit operation at the database of the originating node only upon receipt from at least one of the other nodes of the ready to commit tokens originally sent from the originating node,

    wherein one or more transaction steps or transaction operations have been performed for each transaction being paused,

    and wherein step (b) is performed asynchronously with respect to other ~~transactions~~ transaction steps or transaction operations that are initiated in step (a).

2. (Previously Presented) The method of claim 1 wherein the commit operation in step (b)(v) is performed only upon receipt from each of the one or more other nodes of the ready to commit tokens originally sent from the originating node.

3. (Currently Amended) The method of claim 1 wherein the ~~transactions~~ transaction steps or transaction operations in step (a) are multi-threaded.

4. (Currently Amended) A method of replicating data associated with a plurality of transactions in a data replication system including first and second nodes connected via communication media in a topology, each node including (i) a database, (ii) a replication engine which performs data replication functions, and (iii) an application which executes transactions and posts the transactions to the database, the application being independent of the replication engine, each transaction being one or more transaction steps or transaction operations, including a commit operation, the method comprising:

(a) an application at a first node initiating and performing ~~transactions~~ transaction steps or transaction operations of a transaction to be executed in a database at the first node, the application pausing the commit operation for each transaction being executed in the database at the first node, ~~prior to a commit operation for the transaction~~ wherein one or more transaction steps or transaction operations have been performed for each transaction being paused;

(b) a replication engine at the first node assigning a ready to commit token to the transaction in coordination with the application;

(c) the replication engine at the first node sending the ready to commit token to the second node;

(d) a replication engine at a second node determining whether a target database at the second node is prepared for a commit operation for the transaction corresponding to the ready to commit token, and, if so, sending back the ready to commit token to the first node; and

(e) the application at the first node executing a commit operation at the database at the first node in coordination with the replication engine only upon receipt from the second node of the ready to commit token originally sent from the first node,

wherein the replication engine operates asynchronously with respect to the application until step (b) occurs.

5. (Currently Amended) The method of claim 4 wherein the ~~transactions~~ transaction steps or transaction operations in step (a) are multi-threaded.

6. (Previously Presented) A method of performing dual writes for replicating transactions among plural databases located at different nodes, each transaction being one or more transaction steps or transaction operations, at least some of the transaction steps or transaction operations being update steps or operations, the method comprising:

(a) initiating a transaction at an originating node;

(b) inhibiting the dual write replication process from communicating transaction steps or operations of the transaction with one or more other nodes until an update step or operation occurs within the transaction at the originating node; and

(c) upon the occurrence of the update step or operation and regardless of whether a commit operation has occurred, performing the dual write replication process with respect to the one or more other nodes and sending with the update step or operation all transaction steps or operations for the transaction that have occurred prior to the update step or operation for the transaction, or prior to the previous update step or operation if a previous update step or operation existed for the transaction.

7. (Previously Presented) The method of claim 6 further comprising:

(d) determining if the originating node needs to receive a data record from the one or more other nodes during the dual write replication process; and

(e) sending the data record to the originating node only if it is determined that the originating node needs the data record.

8. (Original) A method of performing dual writes for replicating transactions among plural databases located at different nodes, each transaction being one or more transaction steps or transaction operations, at least some of the transaction steps or transaction operations being

update steps or operations which are performed only after database locking occurs, the method comprising:

(a) initiating a transaction at an originating node; and

(b) the dual write replication process causing database locking to occur at one or more other nodes only upon the occurrence of an update step or operation in the transaction at the originating node.

9. (Previously Presented) The method of claim 8 further comprising:

(d) determining if the originating node needs to receive a data record from the one or more other nodes during the dual write replication process; and

(e) sending the data record to the originating node only if it is determined that the originating node needs the data record.

10. (Previously Presented) The method of claim 1 wherein in step (b)(iii), the originating node sends the ready to commit token to the one or more other nodes.

11. (Previously Presented) The method of claim 1 wherein the commit operation in step (b)(v) is performed only upon receipt from a predetermined number of other nodes of the ready to commit tokens originally sent from the originating node.

12. (Currently Amended) The method of claim 1 ~~wherein each transaction includes one or more transaction steps or operations, and~~ wherein step (b) (iii) is performed by combining the ready to commit token with the last transaction step or operation being replicated prior to a commit operation.

13. (Currently Amended) The method of claim 1 wherein at least one of the nodes is indirectly connected to every other node in the system via at least one other node, and the replicated ~~transactions~~ transaction steps or transaction operations and ready to commit tokens are sent to the indirectly connected nodes via the at least one other node.

14. (Currently Amended) The method of claim 1 wherein each node further includes a table that stores indicia of initiated transactions that are at one or more predesignated intermediate points in the transaction but are not yet committed, and a transaction transmitter which sends selected ~~transactions~~ transaction steps or transaction operations posted to the database and table entries to one or more other nodes, ~~each transaction being one or more transaction steps or transaction operations,~~ the method further comprising:

      (c) entering into the table at the originating node, indicia of the initiated transactions that are at one or more predesignated intermediate points in the transaction but are not yet committed, and assigning a ready to sync token to the transaction at each of the predesignated intermediate points; and the replicating step (b) further includes:

        (vi) the transaction transmitter at the originating node sending the ready to sync tokens in the table of the originating node to the one or more other nodes;

        (vii) determining at each of the one or more receiving nodes whether the database at the receiving node is prepared to properly process the transaction up to the intermediate point associated with the respective ready to sync token, and, if so, the transaction transmitter in each of the other nodes sends back the ready to sync tokens to the respective originating nodes; and

        (viii) stopping the execution of a transaction in the system if the originating node fails to receive back a ready to sync token from at least one of the other nodes in the system for any of the predesignated intermediate points, wherein the transaction continues to execute as long as all ready to sync tokens properly return to the originating node.

15. (Previously Presented) The method of claim 14 further comprising:

      (d) if the execution of a transaction is stopped, restarting the transaction at the beginning of the predesignated intermediate point associated with the ready to sync token that failed to properly return from a node.

16. (Previously Presented) The method of claim 15 wherein the predesignated intermediate points occur at every N transaction steps or operations.

17. (New) The method of claim 1 wherein the transaction is paused prior to the commit operation.

18. (New) The method of claim 4 wherein the transaction is paused prior to the commit operation.